

MULTIMEDIA



UNIVERSITY

STUDENT ID NO

--	--	--	--	--	--	--	--	--	--

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 1, 2019 / 2020 SESSION

DPD 5211 – PROGRAM DESIGN
(DIT Only)

16 OCTOBER 2019
2.30 p.m – 4.30 p.m
(2 Hours)

INSTRUCTIONS TO STUDENT

1. This question paper consists of 8 pages.
2. There are **FOUR (4)** questions. Each question carries 10 marks.
3. Answer ALL questions.
4. Please write your answers in the **Answer Booklet** provided.

STRUCTURED QUESTIONS.

Instruction: Write your answers in the Answer Booklet provided.

QUESTION 1 [10 MARKS]

Teachers of Year 3 from a school in Cyberjaya wanted to do a class arrangement for year 2020. The arrangement is based on the average marks of four subjects (Bahasa Malaysia, English, Mathematics and Science) in end-year examination of 2019 obtained by each student. The requirement is as follows:

Average	Class
70 to 100	Awesome
40 to 69	Brilliant
0 to 39	Champion

Table 1: Class arrangement based on the average marks

- a) **Prepare a flowchart** based on the scenario above and the descriptions below: (5 Marks)
- Initialize all the necessary variables.
 - Get the mark (data type: *float*) for each subject from the user. (You don't have to apply looping for this task).
 - Calculate the total and average mark of the four subjects.
 - Determine the new class for the student based on the requirement shown in Table 1.
 - Display the average mark and class name.
- b) **Write a program** based on the question and your answer in (a). You may refer to the sample output. (5 Marks)

SAMPLE OUTPUT
Enter the mark for: Bahasa Malaysia = 60.5 English = 55 Mathematics = 80 Science = 34.5 Average mark: 57.50 Class: Brilliant

[TOTAL 10 MARKS]
Continued...

QUESTION 2 [10 MARKS]

Table 2 shows the amount of funds collected by the student's representatives for the four areas that were affected by flash flood.

Student Representative \ Area	Jasin	Merlimau	Cheng	Krubong
Kristen	\$200	\$140.50	\$137	\$270
Dave	\$240	\$135	\$123.80	\$90
Sheldon	\$243.60	\$204	\$300	\$132

Table 2: Total amount of funds

Write a program based on the following descriptions:

- Declare a prototype function for the respective functions.
- Create *two global string arrays* named *studentname* (size of 3) and *area* (size of 4). Then assign the student's name and the area name accordingly. Refer to Table 2.
- In *main()*:
 - Create a *two-dimensional array* named *fund* with the appropriate size and assign the values based on the data presented in Table 2.
 - Call function *getTotalFund(...)* and pass the array as the argument.
 - Call function *getHighestFund(...)* and pass the array as the argument.
- In *getTotalFund(...)*:
 - Use *for* loops to calculate the total amount of funds for each student and display it respectively.
 - *Note: This function does not return any value to the main function.*
- In *getHighestFund(...)*:
 - Use *for* loops to find who collected the highest funds and from which area.
 - Display the details.
 - *Note: This function does not return any value to the main function.*

Continued...

You may refer to the following output:

SAMPLE OUTPUT
Total collection from Kristen: RM 747.50
Total collection from Dave: RM 588.80
Total collection from Sheldon: RM 879.60
Sheldon has collected the highest funds - RM 300.00 from area Cheng.

[TOTAL 10 MARKS]

QUESTION 3 [10 MARKS]

Table 3 shows the fare price of a ferry ticket based on the route of departure and destination point.

Route Code	Departure/Destination Name	Fare price (RM)
M	Jetty Merang to Redang Island	55.00
R	Redang Island to Jetty Merang	50.00
K	Jetty Kuala Besut to Perhentian Island	40.00
P	Perhentian Island to Jetty Kuala Besut	35.00

Table 3: Route of ferry and fare amount.

Write a program based on the following descriptions.

- Create a structure called *FerryTicket*. Declare the following variables:
 - Route code: *route_code* (*char*)
 - Number of tickets: *no_pax* (*int*)
 - Ticket price: *fare* (*float*)
 - Subtotal of the tickets price: *subtotal* (*float*)
- Create an array of structure variable called *myTicket* with size of 2, which represents for two users.
- Declare other necessary variables.

Continued...

- Using *for* loop:
 - Get user's input for the route code and number of ticket.
 - Use *switch-case* statement to determine the *fare* based on the route code. Refer to Table 3.
 - Calculate the *subtotal* amount of ticket purchased for each user.
 - Calculate the total amount of the purchased tickets.
 - Display the subtotal amount.
- Display the total amount.

You may refer to the sample output.

SAMPLE OUTPUT
User 1: Enter route code:M Enter no of ticket:1 Subtotal : RM 55.00
User 2: Enter route code:P Enter no of ticket:3 Subtotal : RM 105.00
=Total : RM 160.00

[TOTAL 10 MARKS]

Continued...

QUESTION 4 [10 MARKS]

Program Q4 demonstrates a program that stores customer name, type of laundry service, laundry weight into a text file and calculates the total amount of the service. **Complete the program** based on the instructions below. You don't have to re-write the code that have been provided in *Program Q4*.

The instructions are as follows:

- Declare a prototype function for the respective function.
- Create files variables named *fWrite* and *fRead*.
- Declare all variables as global.
- In *main ()*:
 - A. Open a text file called *customer.txt* for writing.
 - Using a for loop statement to repeat the following process for 2 times (2 customers)
 - Ask the user to enter customer name, type of service and laundry weight
 - B. Write all the data to the text file (refer to Figure 1).
 - C. Close the file.
 - Call function *getTotalAmount()*.
- In *getTotalAmount ()*:
 - D. Open a text file called *customer.txt* for reading.
 - E. Using a *while* loop, read all the data from the file (till the end of the file) and store them in appropriate variables.
 - F. Use *switch* statement to set the service name (*as string*) and the cost per kg based on the service type. Refer to Table 4.

Type of Service	Service	Cost per kg (RM)
1	Wash and folding	8.00
2	Folding	5.00

Table 4: Details of the laundry service

Continued...

- G. Then, use *nested if-else* to calculate the total amount of the service. The formula is:

$$\text{Total} = (\text{cost per kg} \times \text{laundry weight}) + \text{additional charge}$$

Additional charge will be added based on the range of the weight as shown in Table 5.

Laundry weight	Additional charge (RM)
Less or equal to 5	3.00
More than 5 and less than 10	2.00
More than or equal to 10	No additional charge

Table 5: Additional charge

- Display all the data on the command prompt (refer to the sample output).

H. Close the files.

A sample content of *customer.txt* after execution

<customer name> <type of service> <laundry weight>

Richard 2 10.00

Melly 1 5.00

Figure 1: *customer.txt* after execution

SAMPLE OUTPUT

```

Enter customer name: Richard
Enter laundry service: 2
Enter laundry weight: 10
Enter customer name: Melly
Enter laundry service: 1
Enter laundry weight: 5
-----
Customer Name: Richard
Service Name: Folding
Laundry weight: 10.00 kg
Total Amount: RM50.00

Customer Name: Melly
Service Name: Wash and folding
Laundry weight: 5.00 kg
Total Amount: RM43.00

```

} Entered by user

Continued...

Program Q4

```
#include <stdio.h>
#include <string.h>

void getTotalAmount();
FILE *fWrite, *fRead;
char name[20], servicename[20];
int servicetype;
float weight, cost, amount;

void main()
{
    //(A) 1m

    int x;
    for(x=0; x<2; x++)
    {
        fflush(stdin);
        printf("\nEnter customer name: ");
        gets(name);

        printf("Enter laundry service: ");
        scanf("%d", &servicetype);

        printf("Enter laundry weight: ");
        scanf("%f", &weight);

        //(B) 2m
    }

    //(C) 0.25m

    getTotalAmount();
}
```

Continued...


```
//continued from previous page

void getTotalAmount()
{
    //(D) 1m

    printf("\n-----");

    //(E) 2m
    {
        //(F) 1.5m

        //(G) 2m

        printf("\nCustomer Name: %s", name);
        printf("\nService Name: %s", servicename);
        printf("\nLaundry weight: %.2f kg", weight);
        printf("\nTotal Amount: RM%.2f\n", amount);
    }

    //(H) 0.25m
}
```

[TOTAL 10 MARKS]

End of Page.